APPENDIX

Parser Configuration Parameters

BreakAtLine
CookieFieldDelimiters
DateFormat
DisableAnonToKnownUserMapping
DumpCookieFields
DumpQueryArguments
EventDefinition
ExcludedClientIP
ExcludedClientIPRange
ExcludedExtensions
ExcludedSCRange
ExcludedURIPattern
FieldSeparator
GatherStatistics
HeaderLine
IgnoreUserIdsWithSingleOccurrence
ImportID
IncludedExtensions
IncludeFile
IndexedQSArg
IndexedRefQSArg
InputFile
InternalDomainName
KnownUserIdFileName
LogFileFormat
LogicalSiteURIDefinition
MaxLengthCSReferrer
MaxLengthCSURIQuery
MaxLengthCSUserAgent
MaxLengthOutputField
MaxLengthQueryValue
MaxLengthReferrerDomain
MaxLengthReferrerQuery
MaxLengthReferrerURI
MaxLengthURIStem
MaxLengthUserKey
OutputDir
OutputFilePrefix
PageKeyDefinition
ParserThreads
ProcessingLevel
ProcessMaxFileLines
ProcessMaxFiles
ProcessMaxLinesPerDay
ProcessMaxLinesPerSIP
PotentiallyExcludedSCRange

QueryString
QueryStringArg
QueryStringsKeyHashBuckets
RefDomainKeyHashBuckets
ReferrerKeyHashBuckets
RefQueryStringArg
RefUserFieldId
RegdUserDataFile
RegdUserDataHeader
RegdUserIdInCookie
RegdUserIdInQS
RegdUserIdInRefQS
RegdUserIdInUserName
RegdUserIdType // int or string
SecondsSince1970
SessionGapInMinutes
SessionIdFieldInCookie
SessionIdFieldInQS
StripFromQueryString
StripFromRefQueryString
SuccessCodes
URIKeyHashBuckets
URIPairHashBuckets
UserAgentKeyHashBuckets
UserIdFieldInCookie
UserIdFieldInQS
UserIdFieldInRefQS
UserIdPairsFile
UserIdPairsHeader
UserKeyHashBuckets

DETAILS

BREAKATLINE

Purpose
Used for debugging. The Log Parser, while running in debug mode, will execute DebugBreak to give the user an opportunity to debug.
Format
BreakAtLine= <line number>
Default Behavior
Nothing happens.
Valid Example
BreakAtLine= 2345
Invalid Example
BreakAtLine= 2,345 // Hey this is not COBOL! Get rid of that comma
Remarks

COOKIEFIELDDELIMITERS

Purpose
Specifies what characters are used to delimit arg=value pairs in the cookie field.

Format
CookieFieldDelimiters= <string where each character is a delimiter>
Default Behavior
Log Parser uses ";&+" i.e. the three characters ';', '&', and '+' will be used as cookie field delimiters.
Valid Example
CookieFieldDelimiters= ;*&,

Invalid Example
CookieFieldDelimiters= ";*&," // Don't enclose the string in double quotes
Remarks

DATEFORMAT

Purpose
Specifies the date format used in the logs.
Format
DateFormat= <format> // look below for valid formats and how the log parser interprets them
// Pay attention to the last one, SecondsSince1970, which is
// different from all others.
M/D/Y -- Month/Day/Year where Month is 1 to 12, Day is 1 to 31, and Year has four digits.
Mn/D/Y – MonthName/Day/Year where Monthname is 3 or more characters from the name
-- name of the month. E.g. "Jan", "Feb" without the double quotes.
Y/M/D -- Year/Month/Day
Y/Mn/D -- Year/MonthName/Day
D/M/Y -- Day/Month/Year
D/Mn/Y -- Day/MonthName/Year
Y-M-D -- Year-Month-Day
Y-Mn-D-- Year-MonthName-Day
M-D-Y -- Month-Day-Year
Mn-D-Y-- MonthName-Day-Year
D-M-Y -- Day-Month-Year
D-Mn-Y-- Day-MonthName-Year

SecondsSince1970 – Indicates that numbers found in the date column are number of
-- seconds elapsed since 1970.

Default Behavior
When not specified, the year-month-day format is assumed.
Valid Example
DateFormat= Y/M/D // case insensitive
DateFormat= SecondsSince1970
Invalid Example
DateFormat= Y/M-D // The delimiters are mixed ('/' and '-').
Remarks

DISABLEANONTOKNOWNUSERMAPPING

Purpose
Used for debugging.
Format
Default Behavior
Valid Example
Invalid Example

Remarks

DUMPCOOKIEFIELDS

Purpose
Dump all the arg/value pairs in cookies.
Format
Default Behavior
Valid Example
Invalid Example
Remarks

DUMPQUERYARGUMENTS

Purpose
Dump all the arg/value pairs in query strings.
Format
Default Behavior
Valid Example
Invalid Example
Remarks

EVENTDEFINITION

Purpose
Defines an event. Event definitions contain an event name, a file name where those events should be recorded, a set of constraints to be satisfied by a log entry to be considered a match for this event, and an output column that defines the format of the output to be emitted when an event matches a log entry.

The set of constraints to be matched are:

1. Logical site definition – This is a number identified in LogicalSiteURIDefinition. This entry can be empty, in which case a log entry can have any of the known logical site definitions to satisfy this constraint.
2. URI – A log entry should have this URI (or URI fragment) to be considered a match against the event. Note that this URI is not the entire URI, but what is left after the logical site uri match has happened. The URI constraint identifies how the specified URI fragment relates to the log entry's URI. The following three keywords specify how the URI fragment should relate to the log entry's URI.

   {FN} – specifies that the file name portion of the log entry's URI should match the event definition's URI fragment. E.g. {FN}=test.asp indicates that the log entry's URI should end with the file name "test.asp".

   {Prefix} – specifies that the prefix of the log entry's URI should match the event definition's URI fragment.

   {Suffix}

Format
Default Behavior
Valid Example
Invalid Example
Remarks

EXCLUDEDEXTENSIONS

Purpose
Provides a list of comma separated Extensions (including dot). A input log
File with one of these Extensions will be ignored. This allows for a group of
Files to be specified with the understanding that some of them will be Excluded.
Format
Each extension is specified along with the dot. Extensions are comma delimited.
White space is tolerated before and after the comma.
Default Behavior
Valid Example
ExcludedExtensions=.txt, .out, .bin
Invalid Example
ExcludedExtensions=.txt .out, bin
// Two problems: .txt and .out don't have a comma delimiter. bin doesn't have a leading period.
Remarks

EXCLUDEDSCRANGE

Purpose
Provides a pair of comma separated status codes. The parser interprets the first as the start and the second as the end of the range (inclusive) of status codes that will cause a hit to be filtered out.
Format
Start, End
Default Behavior
Valid Example
ExcludedSCRange= 302,304 // 302, 303, and 304 are asked to be excluded
Invalid Example
ExcludedSCRange=302, 300 // Start should be <= End
ExcludedSCRange=302 // Need two params
Remarks

IGNOREUSERIDSWITHSINGLEOCCURRENCE

Purpose
Some customers may issue a new user id for every hit even when the browser indicated that cookies are not being accepted. These extraneously issued user id cookies are essentially useless and should be thrown away. The parser detects that by counting the number of times a user id occurs in the log files (user ids are counted across the entire file – both included and filtered out hits)
Format
IgnoreUserIdsWithSingleOccurrence=Yes
Default Behavior
When this is not specified or IgnoreUserIdsWithSingleOccurrence=No is specified, any user id encountered will be used. The default behavior should be the behavior for most customers.
Valid Example
IgnoreUserIdsWithSingleOccurrence=Yes
Invalid Example
IgnoreUserIdsWithSingleOccurrence= // Need a yes or no
Remarks
If you using this make sure your customer actually needs it. Most customers don't.

IMPORTID

Purpose
Provide an id for the current import.
Format
ImportId=<id number>
Default Behavior
None. This parameter is required.
Valid Example
ImportId=10
Invalid Example
ImportId=test // the id should be a number
Remarks
This is generated by the driver script.

INDEXEDQSARG

Purpose

Directs the parser to extract individual tokens out of a query string value and make those tokens accessible as an array of tokens. This allows a set of values embedded in a query string value to be extracted individually. For example if we have the following query string arg=value pair:

Category=cat1~cat2~cat3

The following will be facilitated when IndexedQSArg is used to tokenize Category using ~ as the delimiter:

Category(1) yields "cat1"
Category(2) yields "cat2"
Category(3) yields "cat3"
Category(4) yields "" // empty string
Category yields "cat1~cat2~cat3"

The query string argument is case insensitive. The index is 1 based(as in Basic).
Note that Category still yields the original string.

Format
IndexedQSArg=<qs arg>, <delimiter>
Default Behavior
When this is not used against a query string arg, you won't be able to access the value as an array.
Valid Example
IndexedQSArg=Category,~ // ~ is the delimiter to tokenize the value of Category
Invalid Example
IndexedQSArg=Category // Delimiter is NOT optional
Remarks

INDEXEDREFQSARG

Purpose
Same as IndexedQSArg but applied to a referrer query string.
Format
Default Behavior
Valid Example

Invalid Example
Remarks

INPUTFILE

Purpose
Provide log File name, one per entry. File names should be fully qualified.
File names can be regular expressions understood by the File system.
Format
InputFile=<Fully qualified File name>. Everything after = until End of line will be considered the File name (leading and trailing white space is Excluded).
Default Behavior
None. This parameter is required.
Valid Example
InputFile=c:\logFiles\2000may1.log
Invalid Example
InputFile=2000may1.log // Not a fully qualified path.
Remarks

INTERNALDOMAINNAME

Purpose
Declares a domain to be a known internal domain. Helps us differentiate between internal and external domains.
Format
InternalDomainName=<domain name>
Default Behavior
If a domain name is not listed as an internal domain name, it will be considered an external domain.
Valid Example
InternalDomainName=www.digimine.com
Invalid Example
Remarks

OUTPUTDIR

Purpose
Provides the name of the Output Directory.
Format
OutputDir=<Fully qualified Directory name>. Everything after = until End of line is considered the Directoryname (leading and trailing white space is Excluded).
Default Behavior
None. This parameter is required.
Valid Example
OutputFile=c:\OutputFiles\
Invalid Example
OutputFile=OutputFiles\ // not a fully qualified path
Remarks

OUTPUTFILEPREFIX

Purpose
Provides a Prefix to be used for all Output Files. The prefix is used as part of

a file name, so use only valid file name chars in the prefix.
Format
OutputFilePRefix=<Prefix>
Default Behavior
None. This parameter is required.
Valid Example
OutputFilePrefix=outFile
Invalid Example
OutputFilePrefix= out|file // | is not a valid file name character
Remarks

USERIDFIELDINCOOKIE

Purpose
Identifies the field name within a Cookie which is used to uniquely identify a visitor to the site. The Parser will use this specification to extract the unique User id from within the Cookie portion of the hit. It is IMPORTANT to note that there can be multiple entries in the config File. The order in which they appear is important. For e.g. the first User id field will be used as the primary User identifier. The second User id will be used as the secondary User identifier. So on and so forth.

> Fields in cookies could have a format such as *SITESERVER=ID1=xyz* and *SITESERVER=ID2=xyz* where there is a group name "SITESERVER" and an identifier of the field that may have small variations (ID1, ID2 etc.) To specify that IDx is the user id field, the specification allows for *token1,delimiterchar* format where token1 should exist and everything after token1 until the delimiterchar is interpreted as part of this id. If there is no need for intervening characters to be skipped (e..g if the id is simply GUID=), then don't specify the delimterchar.

Format
UserIdFieldInCookie=<case insensitive name of the field>
Default Behavior
If this parameter is not specified, no attempt will be made to extract a user id from the cookie field.
Valid Example
UserIdFieldINCookie=UUID=,
UserIdFieldINCookie=UUID=,=
Invalid Example
UserIdFieldINCookie=UUID= // no comma at the end
Remarks
The user id is case sensitive. For. e.g. if the user id exists in the log file as TestCookie, then Testcookie is invalid.

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)_____of_____ were not present for scanning. (Document title)

☐ Page(s)_____of_____ were not present for scanning. (Document title)

☑ ***Scanned copy is best available.*** Drawings fig. 19H, fig 19I, fig. 19J to fig. 19 AE are very dark.